


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)

 Search: ☒ The ACM Digital Library ☐ The Guide

+"edge profiling" +sampling +threshold +transition +hot

SEARCH

THE ACM DIGITAL LIBRARY


[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

 Terms used [edge profiling](#) [sampling](#) [threshold](#) [transition](#) [hot](#)

Found 10 of 166,357

Sort results by

relevance


[Save results to a Binder](#)

Display results

expanded form


[Search Tips](#)
☐ Open results in a new window

[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 10 of 10

 Relevance scale ☐ ☐ ☐ ☐ ☐

### 1 [Rapid profiling via stratified sampling](#)



S. Subramanya Sastry, Rastislav Bodík, James E. Smith

 May 2001 **ACM SIGARCH Computer Architecture News , Proceedings of the 28th annual international symposium on Computer architecture ISCA '01**, Volume 29 Issue 2

Publisher: ACM Press

Full text available: pdf(1.02 MB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

*Sophisticated binary translators and dynamic optimizers demand a program profiler with low overhead, high accuracy, and the ability to collect a variety of profile types. A profiling scheme that achieves these goals is proposed. Conceptually, the hardware compresses a stream of profile data by counting identical events; the compressed profile data is passed to software for analysis. Compressing the high-bandwidth event stream greatly reduces software overhead. Because optimizations can tolerate ...*

### 2 [Online feedback-directed optimization of Java](#)



Matthew Arnold, Michael Hind, Barbara G. Ryder

 November 2002 **ACM SIGPLAN Notices , Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '02**, Volume 37 Issue 11

Publisher: ACM Press

Full text available: pdf(463.00 KB)

 Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes the implementation of an online feedback-directed optimization system. The system is fully automatic; it requires no prior (offline) profiling run. It uses a previously developed low-overhead instrumentation sampling framework to collect control flow graph edge profiles. This profile information is used to drive several traditional optimizations, as well as a novel algorithm for performing feedback-directed control flow graph node splitting. We empirically evaluate this system ...

**Keywords:** adaptive optimization, dynamic optimization, online algorithms, virtual machines

### 3 [Edge profiling versus path profiling: the showdown](#)



Thomas Ball, Peter Mataga, Mooly Sagiv

 January 1998 **Proceedings of the 25th ACM SIGPLAN-SIGACT symposium on**



**Principles of programming languages****Publisher:** ACM PressFull text available:  pdf(1.84 MB)Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**4** Design and evaluation of dynamic optimizations for a Java just-in-time compilerToshio Suganuma, Toshiaki Yasue, Motohiro Kawahito, Hideaki Komatsu, Toshio Nakatani  
July 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 27 Issue 4**Publisher:** ACM PressFull text available:  pdf(1.60 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The high performance implementation of Java Virtual Machines (JVM) and Just-In-Time (JIT) compilers is directed toward employing a dynamic compilation system on the basis of online runtime profile information. The trade-off between the compilation overhead and performance benefit is a crucial issue for such a system. This article describes the design and implementation of a dynamic optimization framework in a production-level Java JIT compiler, together with two techniques for profile-directed o ...

**Keywords:** JIT compiler, Recompilation, adaptive optimization, code specialization, dynamic compilation, profile-directed method inlining

**5** Software profiling for hot path prediction: less is moreEvelyn Duesterwald, Vasanth Bala  
November 2000 **ACM SIGOPS Operating Systems Review , ACM SIGARCH Computer Architecture News , Proceedings of the ninth international conference on Architectural support for programming languages and operating systems ASPLOS-IX**, Volume 34 , 28 Issue 5 , 5**Publisher:** ACM PressFull text available:  pdf(286.07 KB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

**6** Software profiling for hot path prediction: less is moreEvelyn Duesterwald, Vasanth Bala  
November 2000 **ACM SIGPLAN Notices**, Volume 35 Issue 11**Publisher:** ACM PressFull text available:  pdf(2.43 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Recently, there has been a growing interest in exploiting profile information in adaptive systems such as just-in-time compilers, dynamic optimizers and, binary translators. In this paper, we show that sophisticated software profiling schemes that provide highly accurate information in an offline setting are ill-suited for these dynamic code generation systems. We experimentally demonstrate that hot path predictions must be made early in order to control the rising cost of missed opportunity tha ...

**7** Continuous program optimization: A case studyThomas Kistler, Michael Franz  
July 2003 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,  
Volume 25 Issue 4



**Publisher:** ACM Press

Full text available:  pdf(877.67 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

Much of the software in everyday operation is not making optimal use of the hardware on which it actually runs. Among the reasons for this discrepancy are hardware/software mismatches, modularization overheads introduced by software engineering considerations, and the inability of systems to adapt to users' behaviors. A solution to these problems is to delay code generation until load time. This is the earliest point at which a piece of software can be fine-tuned to the actual capabilities of the ...

**Keywords:** Dynamic code generation, continuous program optimization, dynamic reoptimization

## 8 Optimising hot paths in a dynamic binary translator



David Ung, Cristina Cifuentes

March 2001 **ACM SIGARCH Computer Architecture News**, Volume 29 Issue 1

**Publisher:** ACM Press

Full text available:  pdf(890.10 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In dynamic binary translation, code is translated "on the fly" at run-time, while the user perceives ordinary execution of the program on the target machine. Code fragments that are frequently executed follow the same sequence of flow control over a period of time. These fragments form a hot path and are optimised to improve the overall performance of the program. Multiple hot paths may also exist in programs. A program may choose to execute in one hot path for some time, but later switch to another ...

**Keywords:** binary translation, dynamic compilation, dynamic execution, run-time profiling

## 9 Targeted Path Profiling: Lower Overhead Path Profiling for Staged Dynamic Optimization Systems

Rahul Joshi, Michael D. Bond, Craig Zilles

March 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04**

**Publisher:** IEEE Computer Society

Full text available:  pdf(281.40 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

In this paper, we present a technique for reducing the overhead of collecting path profiles in the context of a dynamic optimizer. The key idea to our approach, called Targeted Path Profiling (TPP), is to use an edge profile to simplify the collection of a path profile. This notion of profile-guided profiling is a natural fit for dynamic optimizers, which typically optimize the code in a series of stages. TPP is an extension to the Ball-Larus Efficient Path Profiling algorithm. Its increased efficiency ...


## 10 Overcoming the challenges to feedback-directed optimization (Keynote Talk)



Michael D. Smith

January 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN workshop on Dynamic and adaptive compilation and optimization DYNAMO '00**, Volume 35 Issue 7

**Publisher:** ACM Press

Full text available:  pdf(1.33 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

*Feedback-directed optimization (FDO) is a general term used to describe any technique that alters a program's execution based on tendencies observed in its present or past*



*runs. This paper reviews the current state of affairs in FDO and discusses the challenges inhibiting further acceptance of these techniques. It also argues that current trends in hardware and software technology have resulted in an execution environment where immutable executables and traditional static optimizations are ...*

Results 1 - 10 of 10

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.  
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)